IoT Week 2022

The IoTAC Software Security-by-Design Platform

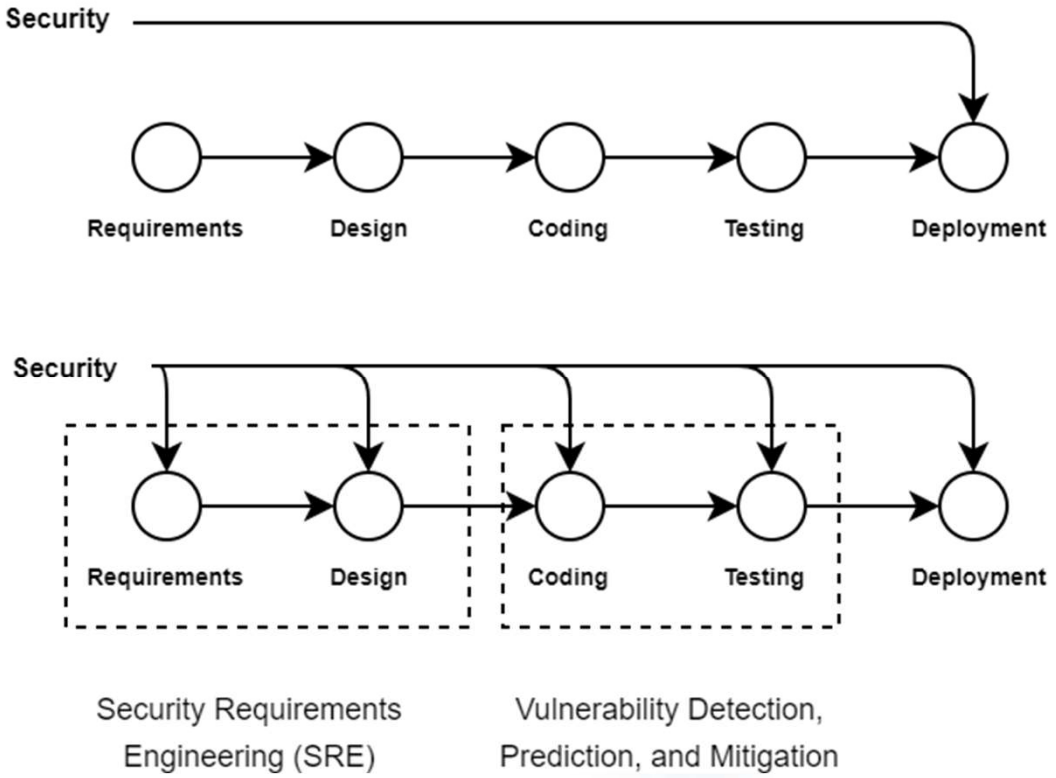
Miltiadis Siavvas,

Centre for Research and Technology Hellas

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952684.

The Security-by-Design Paradigm

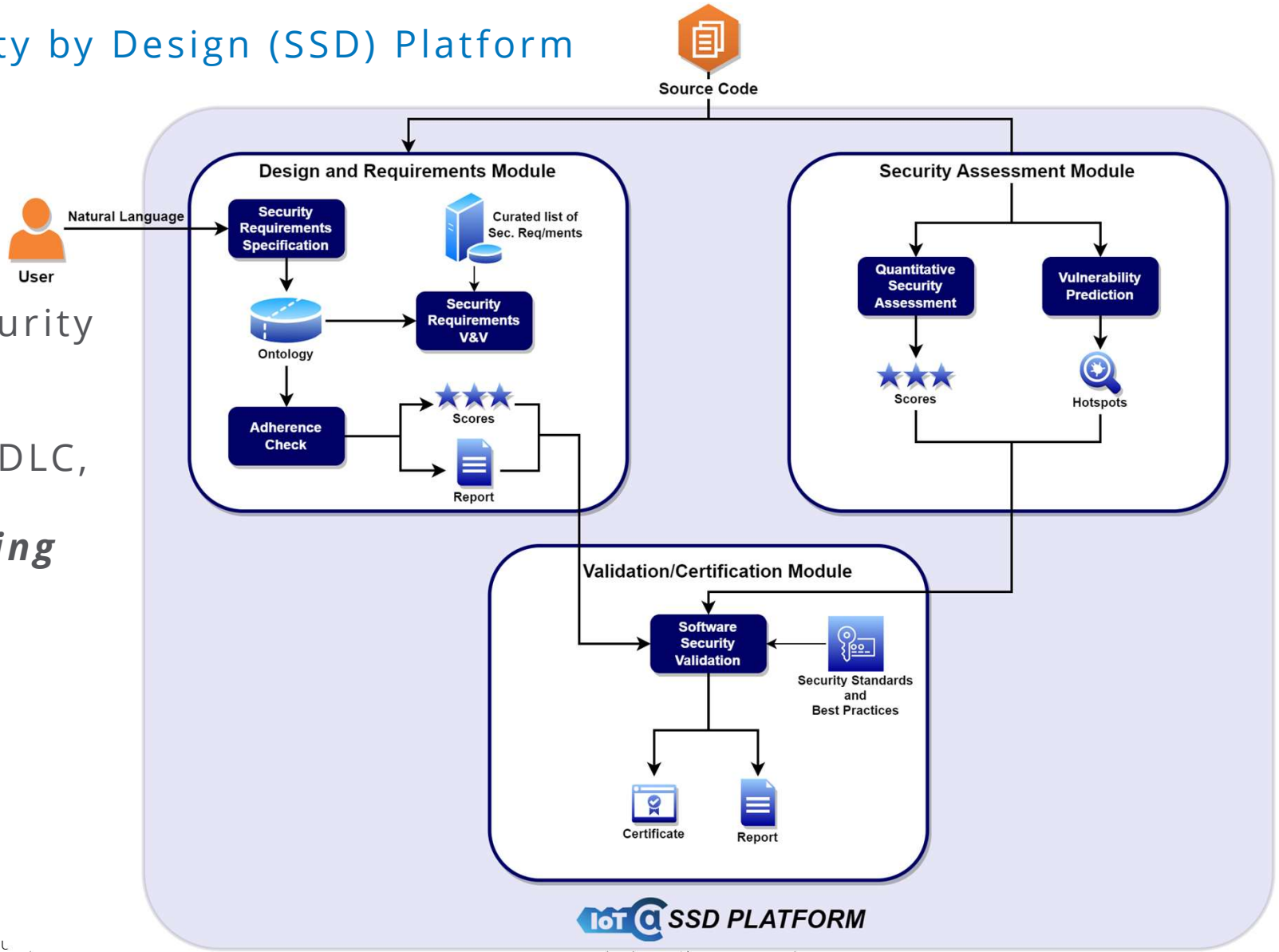
- Securing only the architecture of an IoT system is not enough
- Vulnerabilities in software that is running on IoT systems may infringe their security (weakest link principle)
- Traditionally, Security is treated as an afterthought during the overall Software Development Lifecycle (SLDC)
- In the Secure SDLC, Security is treaded during the various phases of the design and development



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952684.

Software Security by Design (SSD) Platform

- Provide novel solutions for ***monitoring*** and ***optimizing*** the security of IoT Software Applications throughout their SDLC, and **validating/*certifying*** their security level



This project has received funding from the EU 2020 research and innovation programme under grant agreement No 952684.

Access Control

Design and Requirements Module

Problem:

- A significant portion of software vulnerabilities are introduced during the design and requirements phase, mainly due to (i) incorrect, (ii) unclear, and (iii) missing security requirements.

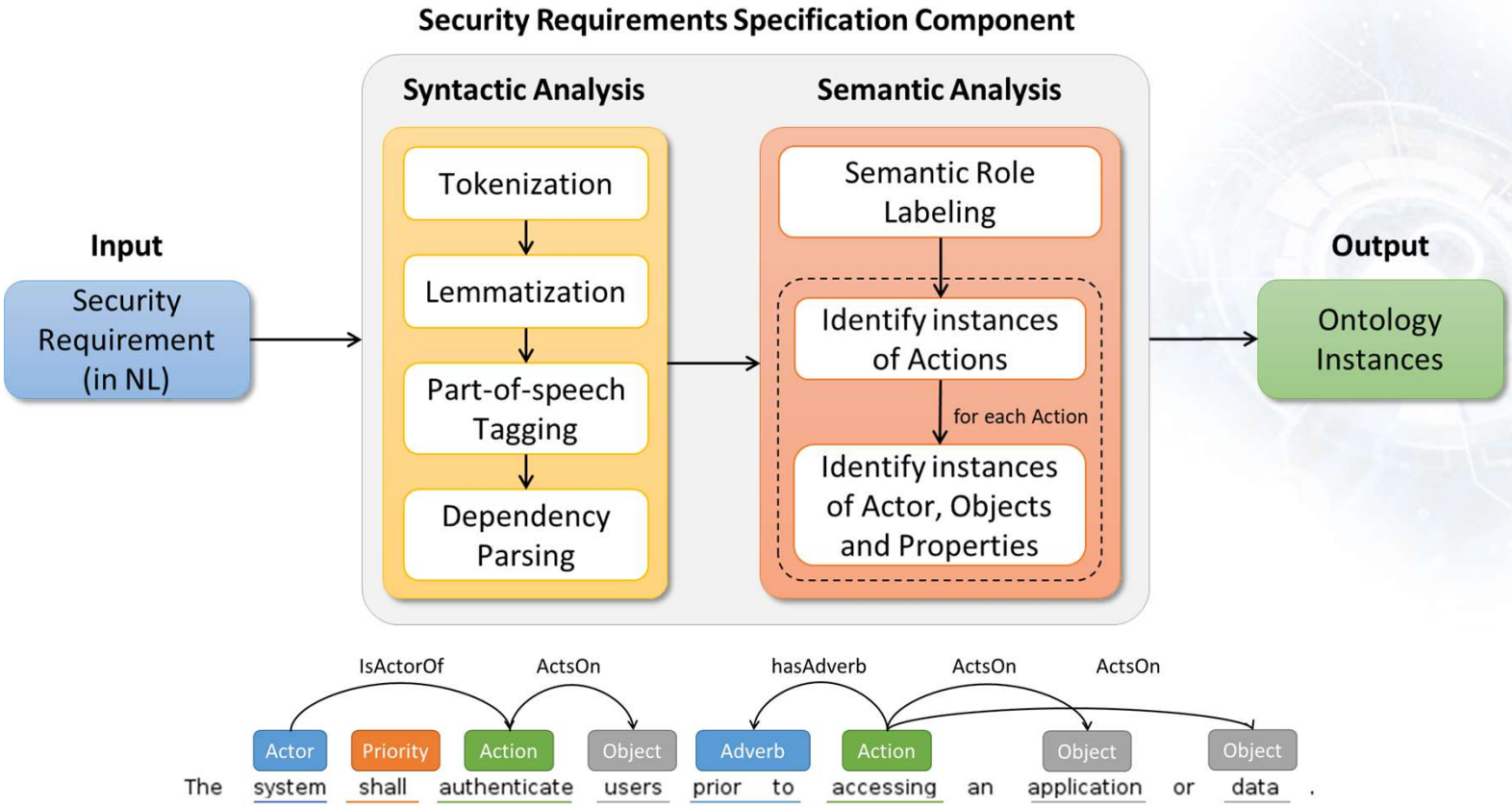
Purpose:

- Propose novel methods and techniques that will
  - Facilitate the correct specification of Software Security Requirements
  - Verify and Validate the defined Software Security Requirements and recommend potential refinements and/or additions
  - Check the adherence of the final software applications to the originally imposed security requirements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952684.

Software Security Requirements Specification (SSRS)

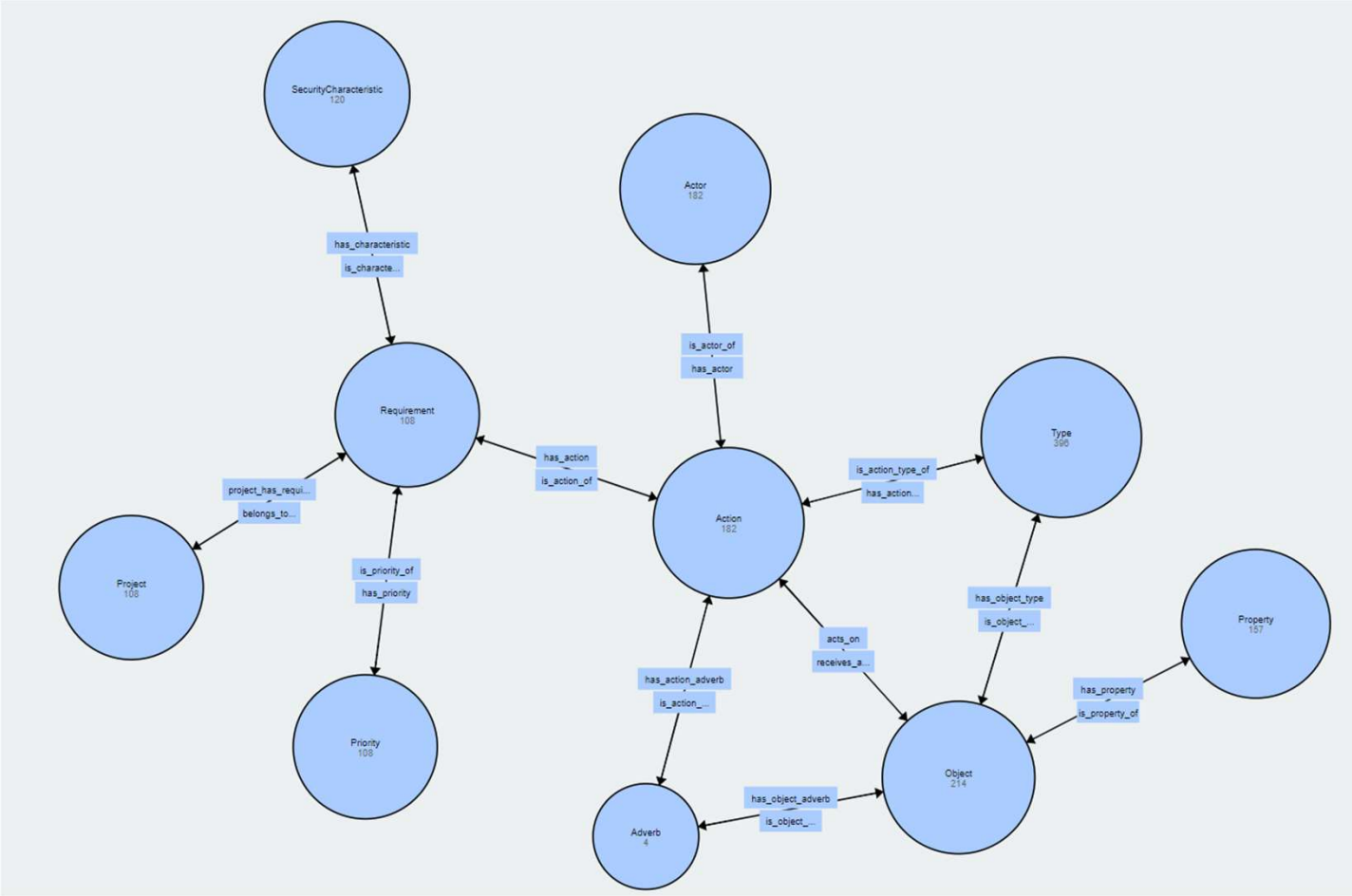
- A mechanisms for facilitating the definition of software security requirements
- It receives Software Security Requirements expressed in natural language (i.e., pure text), automatically identifies its main concepts (e.g., actor, action, etc.), and presents them in a unified form (i.e., ontology instances).



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952684.

Software Security Requirements Specification (SSRS)

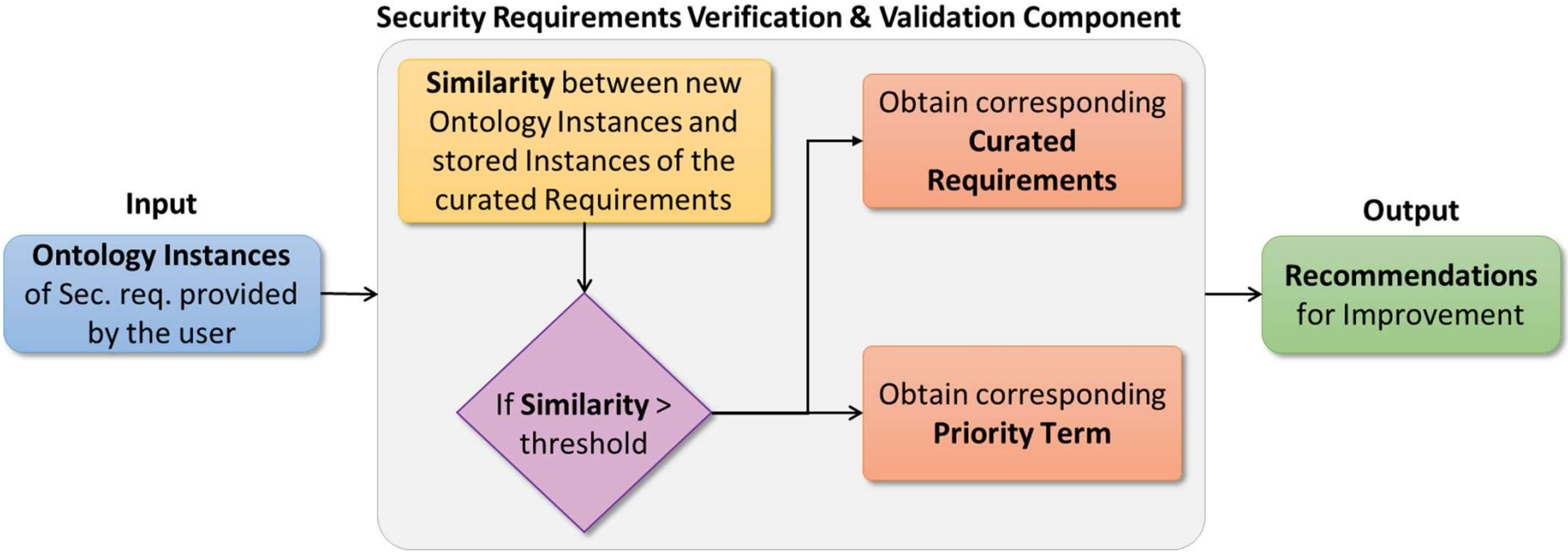
- Software Security Requirements Ontology:
  - An Ontology for storing Software Security Requirements has been set up
  - A schema has been devised based on our research
  - 110 curated software security requirements collected (up to now)



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952684.

Software Security Requirements Verification and Validation (SSRVV)

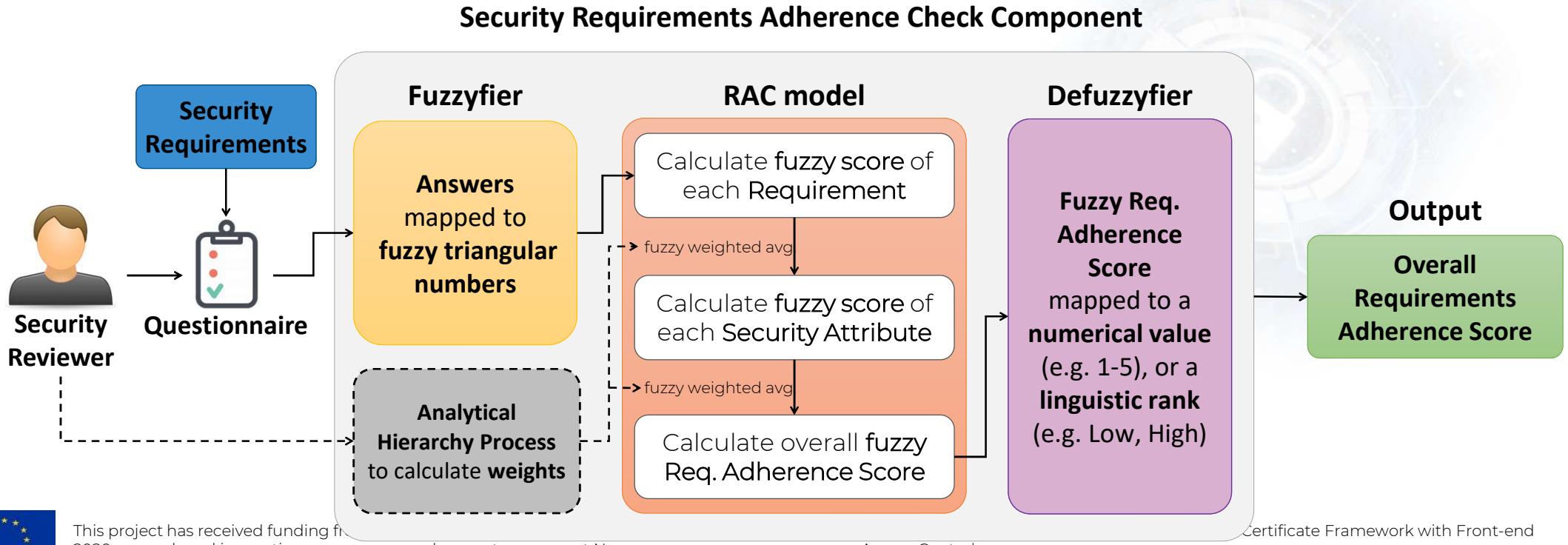
- A mechanism for verifying and validating the correctness and completeness of the defined Software Security Requirements, as well as for recommending potential refinements
- It receives the user-defined Software Security Requirements (defined through the SSRS) from the Ontology, assesses their correctness and completeness, and provides refinement recommendations based on a curated list of well-defined software security requirements.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952684.

Software Security Requirements Adherence Check (SSRAC)

- A methodology for evaluating (in a quantifiable way) the extent to which a given software product adheres to its originally imposed security requirements
  - The approach is based on expert judgements
  - It is based on a hierarchical meta-model and encompasses concepts from multi-criteria decision making field
  - Fuzzy logic is employed in order to model uncertainty



This project has received funding f
2020 research and innovation programme under grant agreement No
952684.
Certificate Framework with Front-end
Access Control
https://iotac.eu

Software Security Assessment Module

Problem:

- The vast majority of software vulnerabilities are caused by a limited number of security flaws (typically programming faults) introduced by developers during the coding phase
- There is a need for mechanisms and tools that will help project managers and developers know the security level of their code, detect security issues, and remove them prior to the product release.

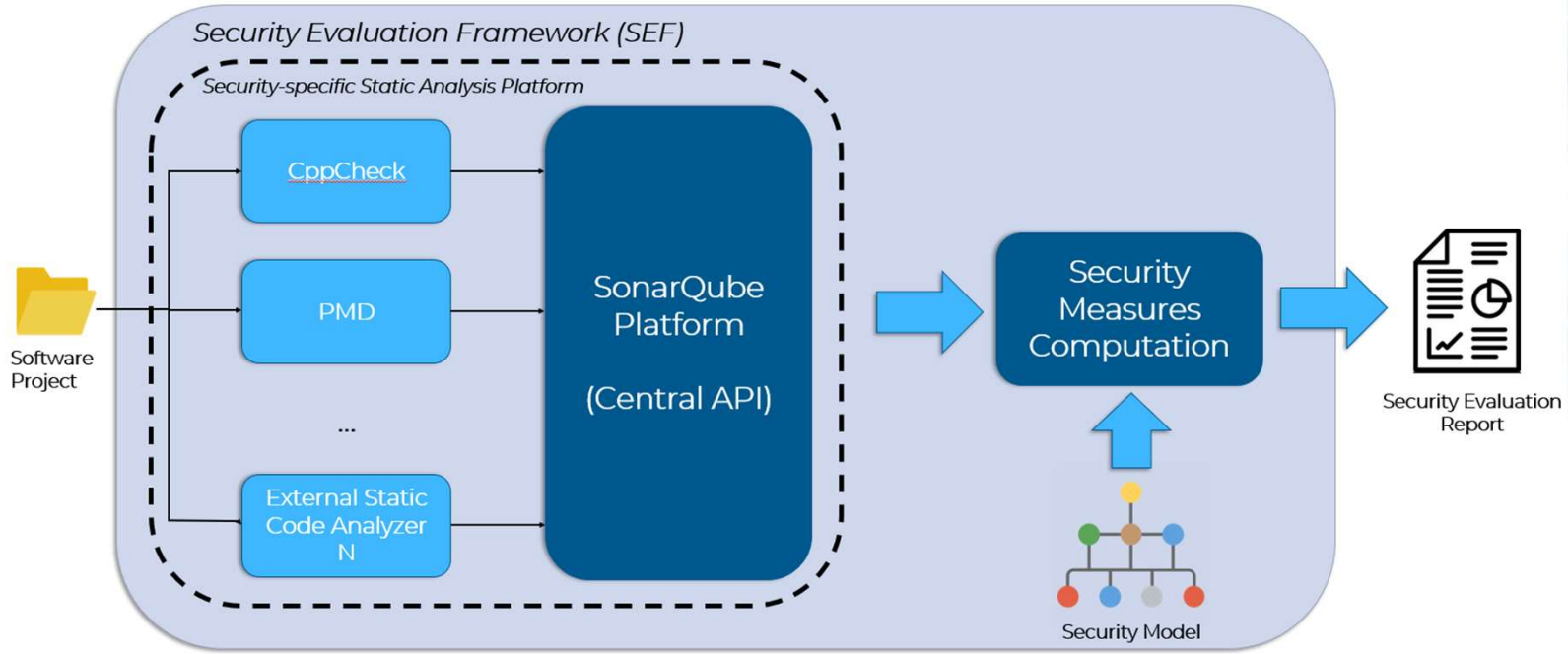
Purpose:

- Propose novel methods and techniques that will
  - Provide high-level measures that will reflect the security level of the source code
  - Highlight security hotspots, i.e., components that are likely to be vulnerable
  - Detect code-level issues that may correspond to actual vulnerabilities

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952684.

Security Evaluation Framework (SEF)

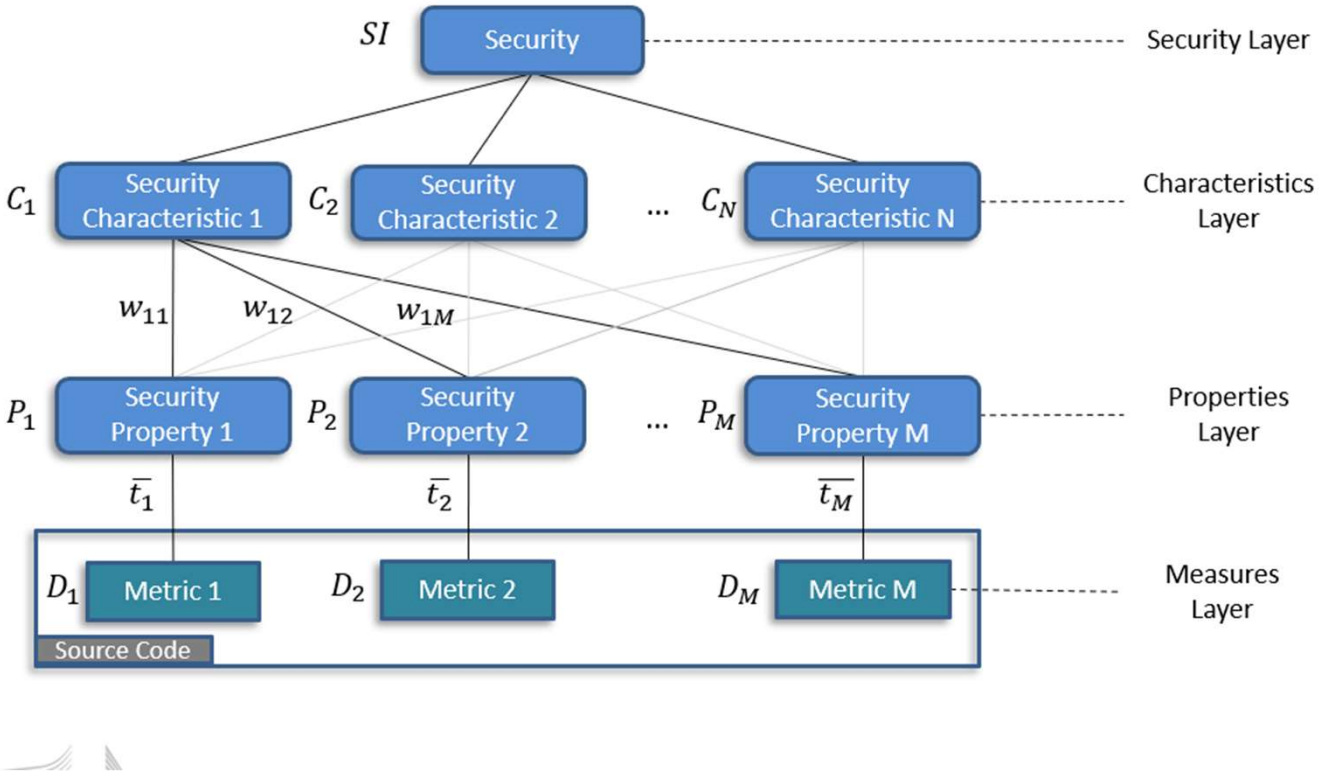
- A mechanism that evaluates the security level of the source code of a given IoT software application in a quantitative manner, based on the results of security-specific **static analysis**.
- Calculates **high-level security measures** by aggregating the low-level results of the security-specific static analysis.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952684.

Security Evaluation Framework (SEF)

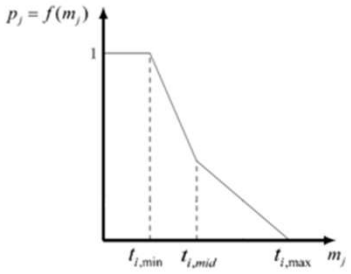
- Typical Structure of a Security Model used by SEF



$$SI = \frac{1}{N}\sum_{i=1}^{N} C_i$$

$$C_i = \sum_{j=1}^{M} w_{ij} P_j$$

$$P_j = f(D_j)$$

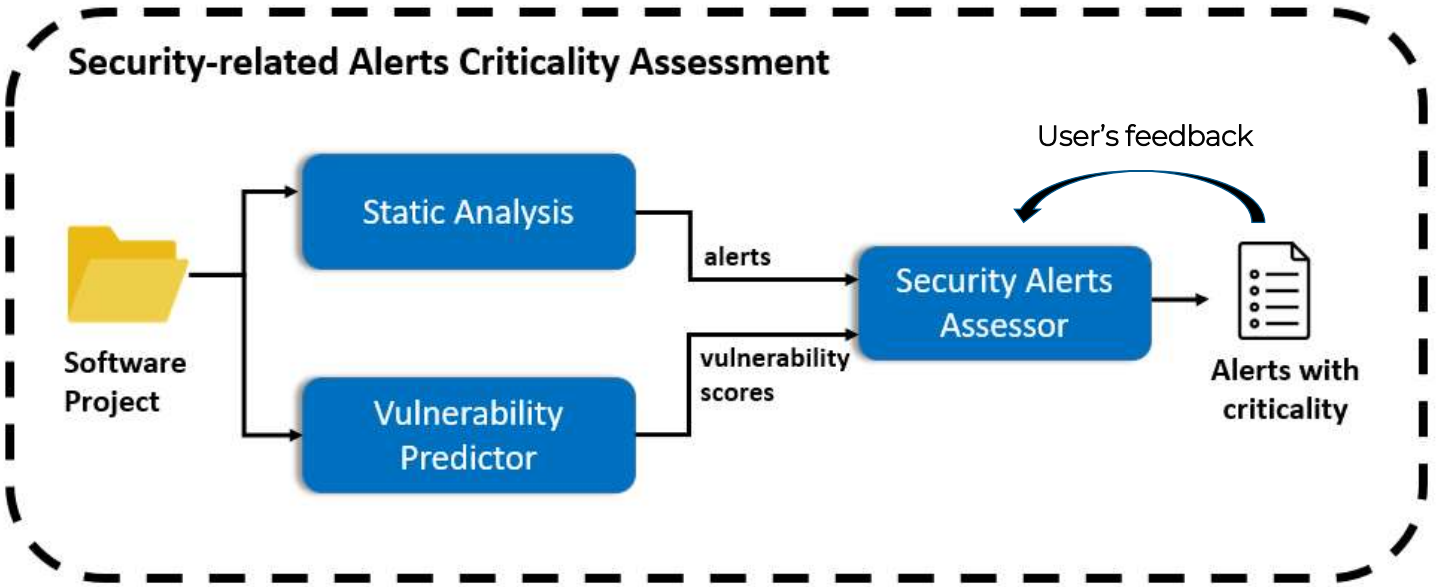


$$\overline{t_i} = [t_{i,min} \quad t_{i,mid} \quad t_{i,max}]$$

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952684.

Security-related Alerts Criticality Assessment (SACA)

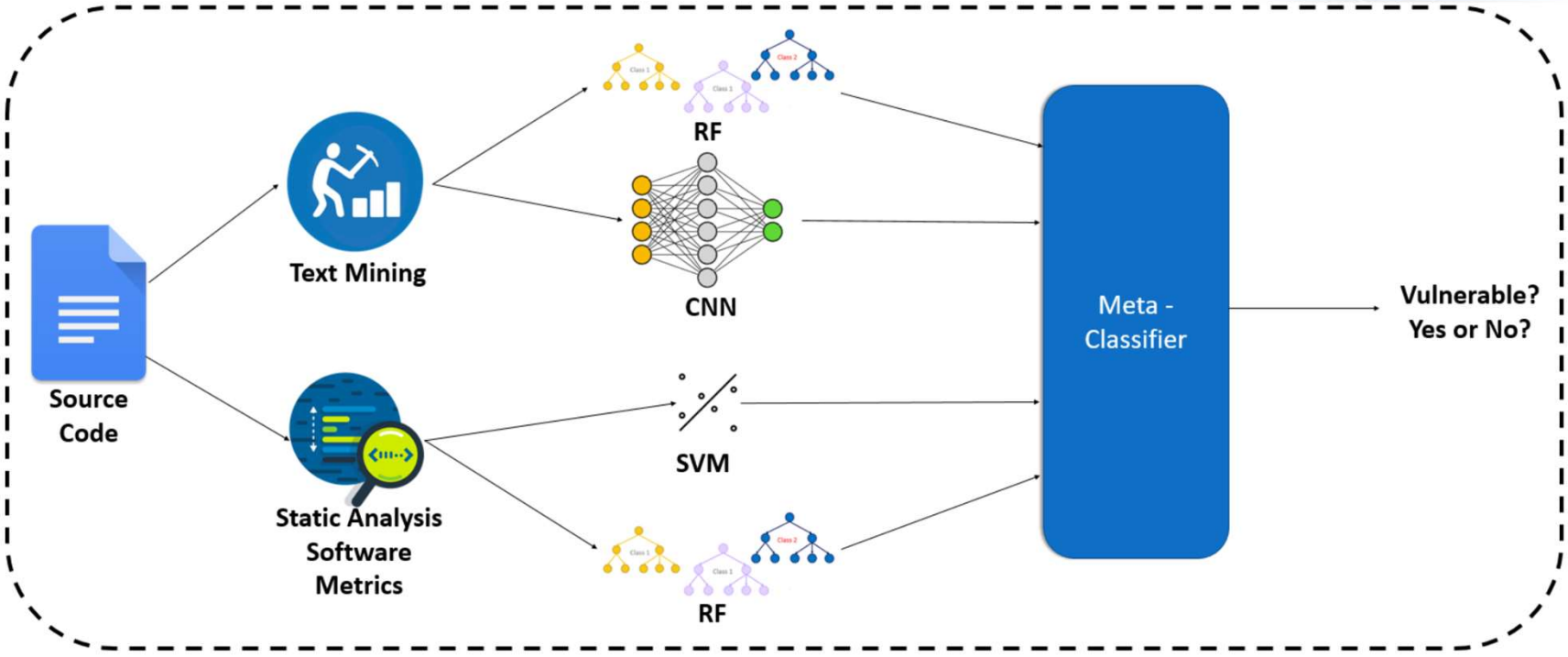
- A Multi-layer Perceptron (MLP) that assesses the criticality of security-related alerts based on information retrieved from the alerts themselves, vulnerability prediction, and user feedback.
- The model can be retrained based on changes provided by the developers



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952684.

Vulnerability Prediction Models (VPMs)

- The purpose of Vulnerability Prediction Models is to detect security hotspots, i.e., software components that are likely to contain security vulnerabilities in order to help developers to prioritize inspection efforts
- **Main Focus:** text mining, software metrics, and combination



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952684.

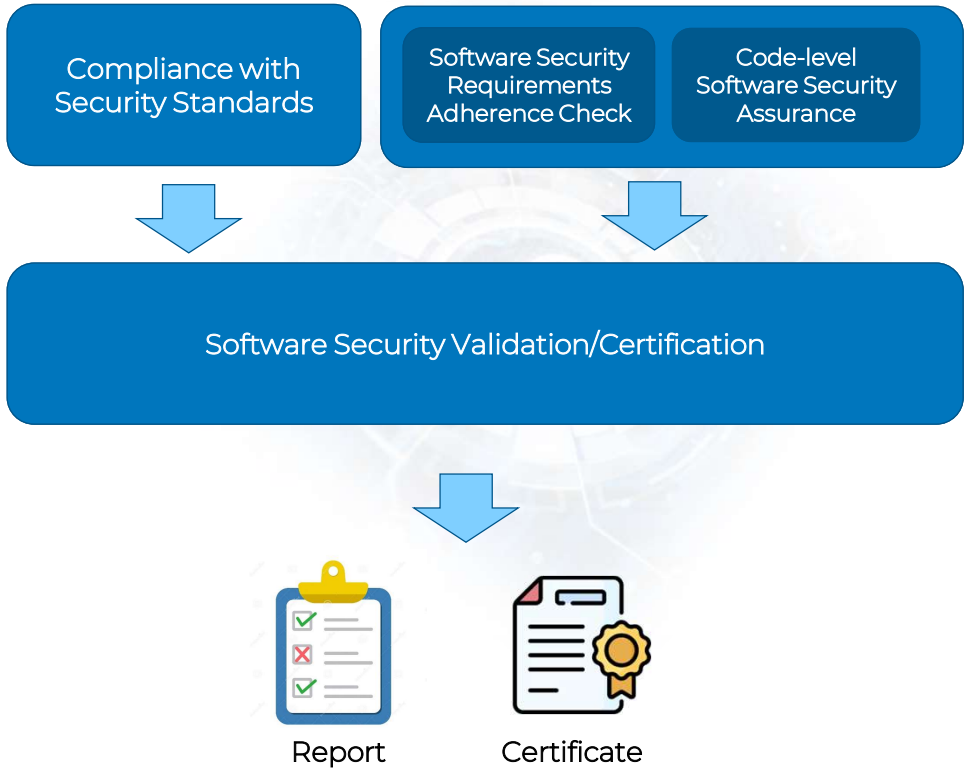
Software Security Validation/Certification

Purpose:

- Propose novel methods for security certification (in fact, security validation) of software applications running on IoT Platforms.
- Investigation of which parts of the overall process that can be automated

Solution:

- A unified methodology that will incorporate security information:
  - From the product itself (through dedicated analyses)
  - From popular Security Standards



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952684.

Software Security Validation/Certification

Security Standards – Requirement Extraction - Example:

using a [*selection: digital signature mechanism, published hash*] prior to installing those updates.

3.2.4 Secure Update of Application

Requirement

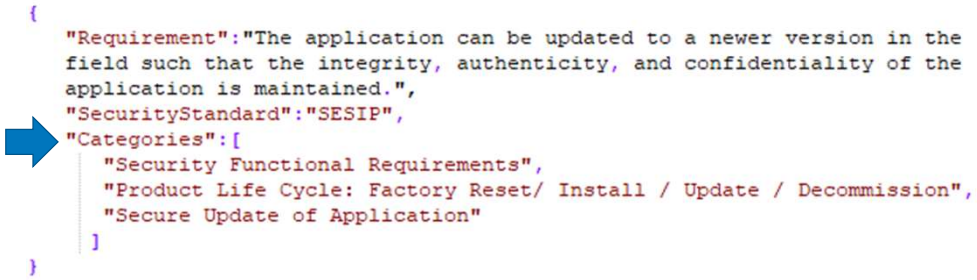
The application can be updated to a newer version in the field such that the integrity, authenticity, and confidentiality of the application is maintained.

Value

Addressing security flaws, functional bugs, or improvements may require an update of the application in the field. Composite developers and evaluators can be assured that the application is not disclosed or inappropriately modified during this update.

Consideration

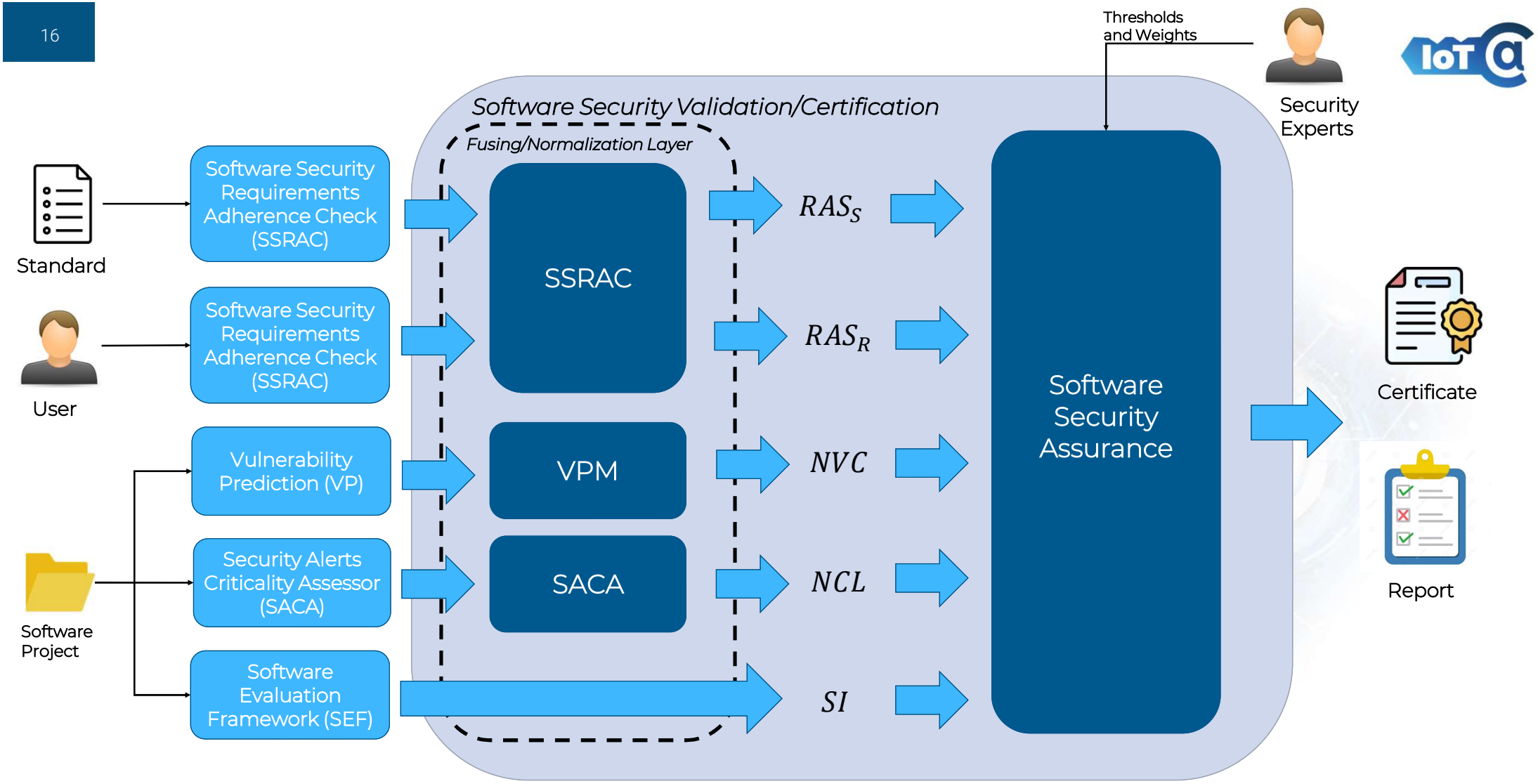
The update mechanism shall counter downgrade attacks ("updating" to an older, potentially more vulnerable



```
{
  "Requirement":"The application can be updated to a newer version in the
  field such that the integrity, authenticity, and confidentiality of the
  application is maintained.",
  "SecurityStandard":"SESIP",
  "Categories":[
    "Security Functional Requirements",
    "Product Life Cycle: Factory Reset/ Install / Update / Decommission",
    "Secure Update of Application"
  ]
}
```

- The Software Security Requirements Specification (SSRS) Mechanism of the Design and Requirements Module is leveraged for the construction of the dedicated ontology

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952684.

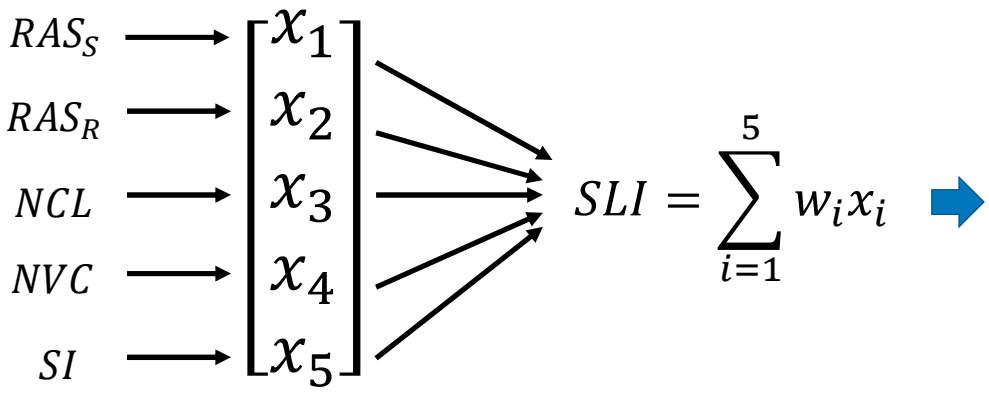


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952684.

Software Security Validation/Certification

Security Level Indicator:

- The normalized outputs of the selected components of the SSD Platform are treated as the Criteria of the validation/certification procedure
- They are aggregated in order to produce the final indicator, the Security Level Indicator (SLI)



Category	Name	Score
A	Outstanding	>= 90%
B	Good	>= 80%
C	Fairly Good	>= 70%
D	Moderate	>= 50%
E	Weak	>= 40%
F	Poor	< 40%

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952684.

ETSI IoT Week 2022

THANK YOU!

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 952684.

IoTAC - Security By Design IoT Development and Certificate Framework with Front-end Access Control

https://iotac.eu